Goddard Mission Services Evolution Center (GMSEC) Reference Architecture

Dan Smith
Dan.Smith@nasa.gov

John Bristow
John.O.Bristow@nasa.gov

GMSEC Background

GMSEC was established in 2001 to coordinate ground and flight data systems development and services at GSFC

- Goals
  - Simplify integration and development
  - Facilitate technology infusion over time
  - Support evolving operational concepts
  - Avoid vendor lock-in
  - Allow for mix of heritage and new components
- Concepts
  - Standardize interfaces – not components
  - Provide a middleware infrastructure
  - Allow users to choose – GMSEC doesn't decide which components are best or dictate which components a mission must use. It's the mission/user's choice!
- Funding – Earth and Space Science OPS, Exploration, some missions

GMSEC Lab

Visit us at our new location:

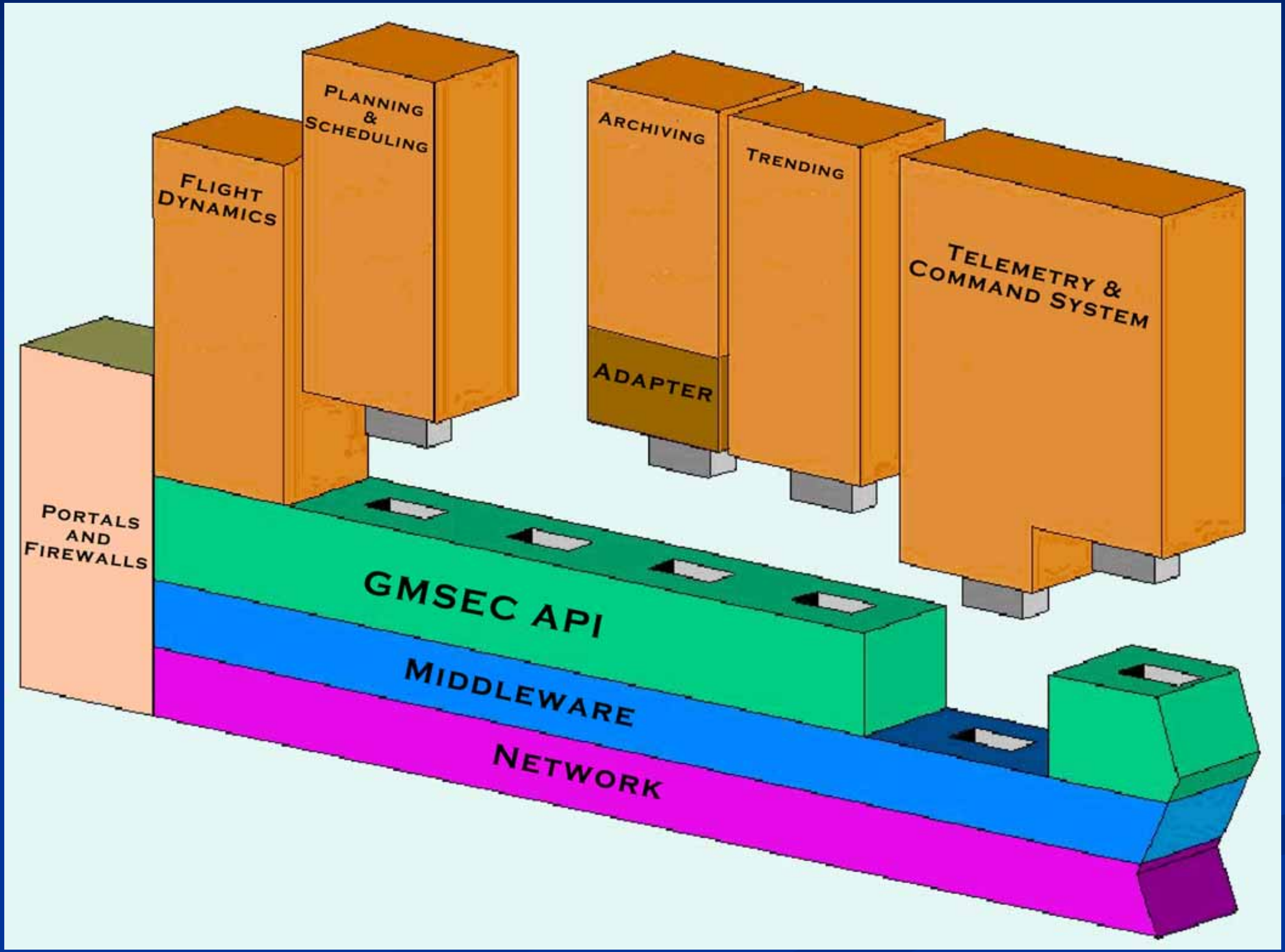
GSFC, Bldg 23
Room N307

See GMSEC Labs at other NASA Centers as well:

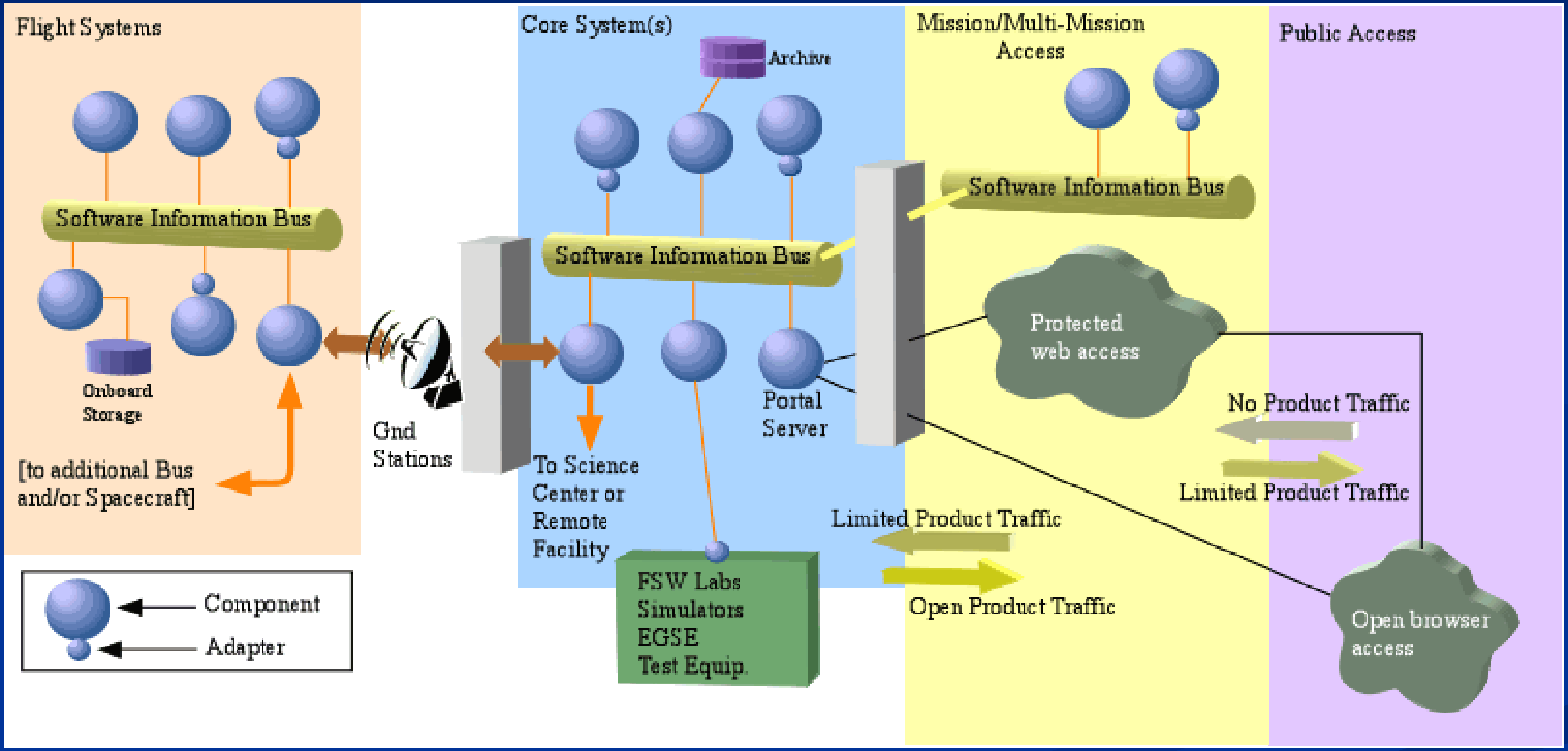
KSC – now open
JSC, MSFC – shared with other labs

Plug-and-Play Concept

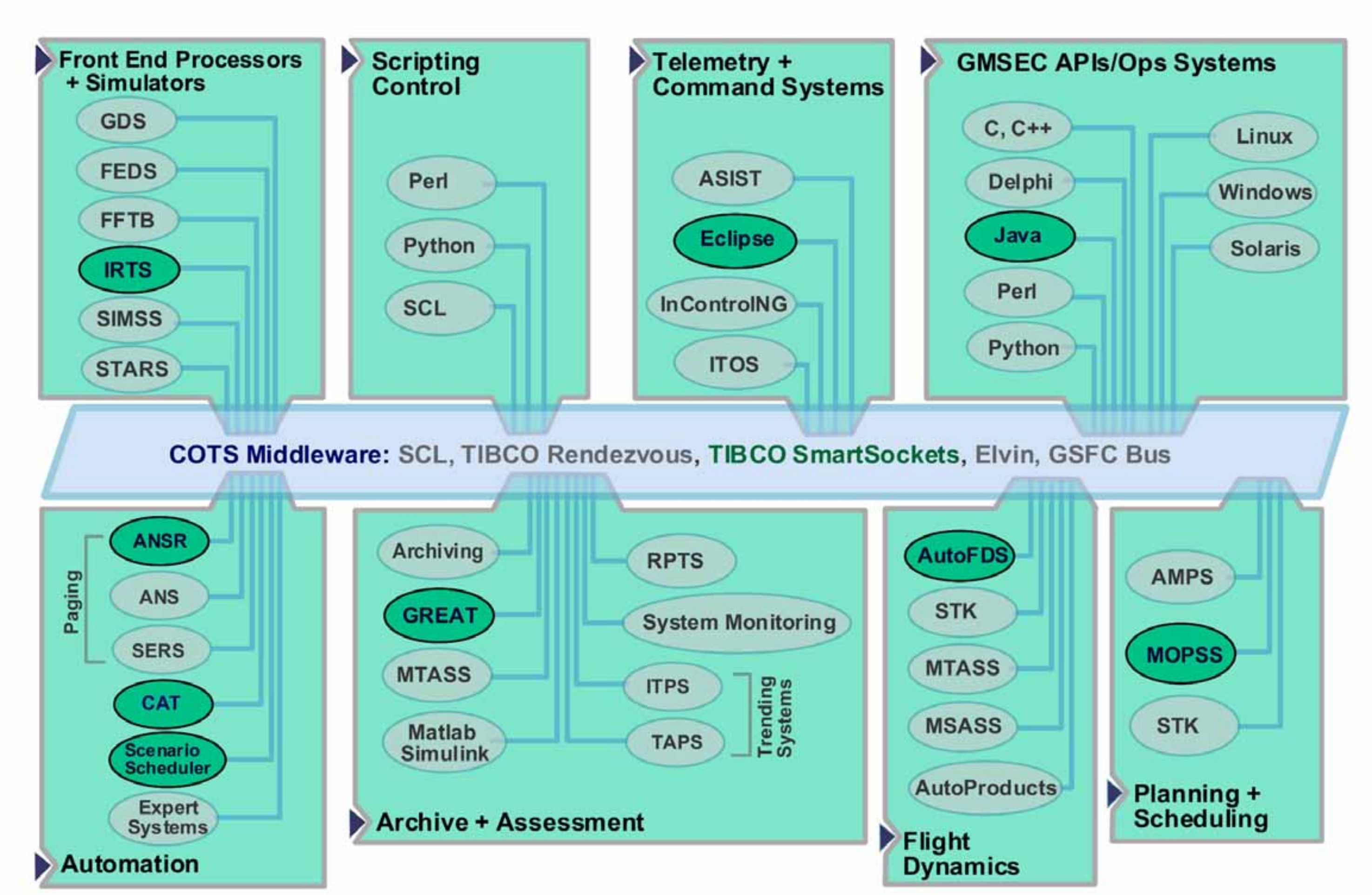
By creating a "framework", individual applications can be easily integrated into an existing system without regard to many underlying implementation details.



GMSEC Architecture



GMSEC Component Catalog



Choices are available for many subsystems. The TRMM mission selected catalog components to best meet their reengineering needs.

GMSEC System Status

- Dozens of components now available
- Automated test package for over 8,000 combinations of platforms, OS, languages, middleware
- Full documentation set
- GMSEC Architecture Application Programming Interface (API)
  - Middleware Independent – avoiding vendor lock-in of critical framework
    - Many middleware options including: TIBCO SmartSockets, TIBCO Rendezvous, ICS SWB, GSFC Bus, IBM WebSphere
    - Middleware can be changed at runtime
  - Standard API and message formats
  - Cross Platform API
    - Linux RH (7.1-7.3,9, + Enterprise), Suse (8.1)
    - Windows NT/2000/XP
    - Solaris (gcc + cc)
  - Multiple Languages
    - C, C++, Java, Perl, Python, Delphi
    - Interfaces for Matlab, Satellite Toookit

GMSEC Operational Status

Existing mission reengineering efforts

- Tropical Rainforest Measuring Mission (TRMM)
  - Goal: reduce operations cost by 50%
  - Pathfinder for Terra, Aqua, Aura automation (2005-2006)
- Small Explorer (SMEX) missions – SWAS, WIRE, TRACE, SAMPEX
  - Pathfinder for low-cost fleet operations & updating existing space science missions

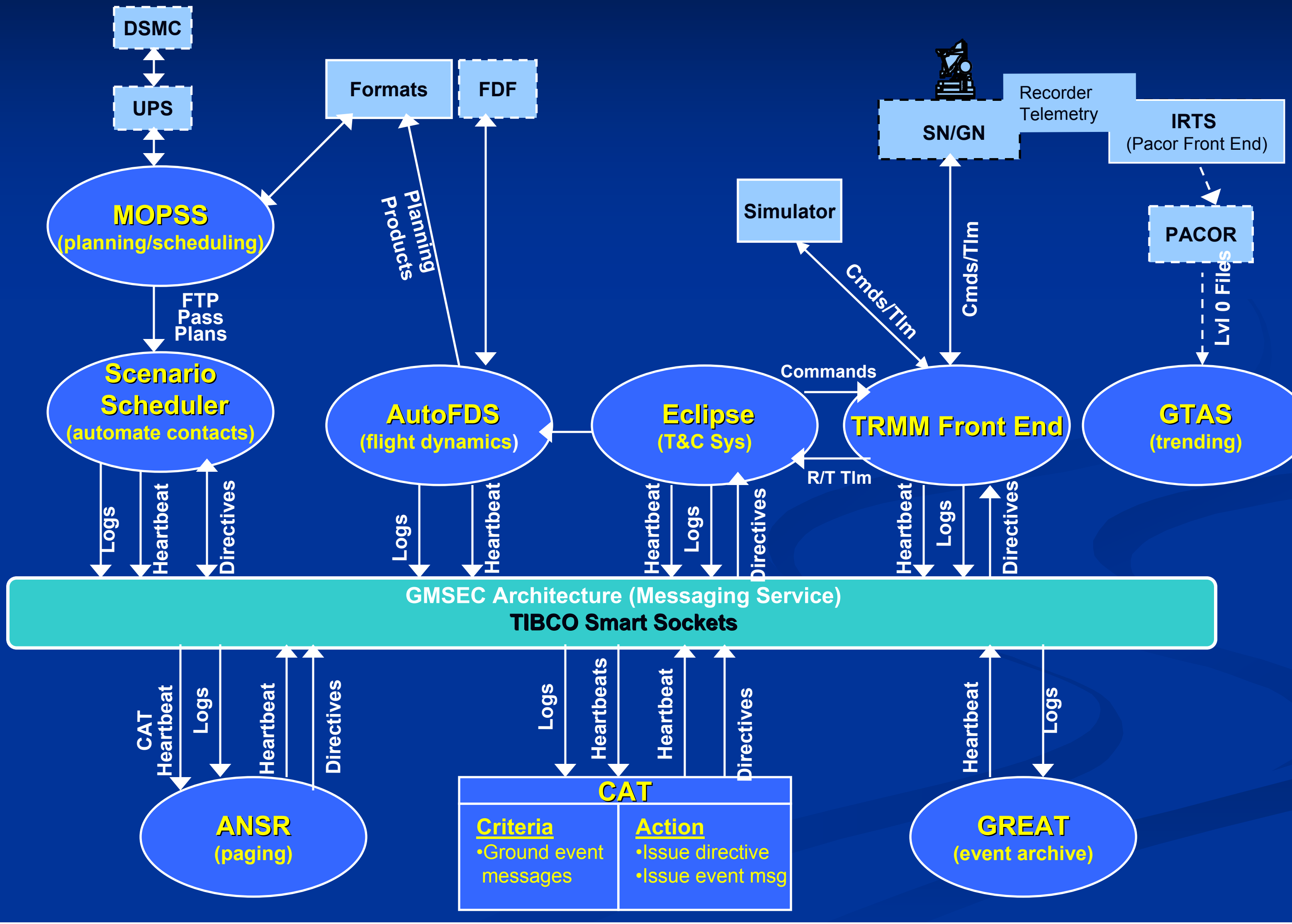
New GSFC missions

- ST5 Constellation system in pre-launch test
- Co-developing: MMS, GPM (long-range schedules)
- Discussions/planning: LRO, GLAST, SDO
- Sharing tools and approach: JWST

Other applications

- Other NASA Centers and related facilities: GSFC/WFF, JPL, JHU/APL, MSFC, JSC, KSC, overall Exploration Initiative
- GSFC Flight Dynamics Facility (FDF)
- Commercialization interest: Several major commercial satellite operators

GMSEC System Example TRMM Logical Architecture



New Capabilities Enabled

System-Wide Automation

- Monitor bus traffic, take action based on defined rules
- Crosses boundaries of all component domains

Entire System Monitoring

- Subscribe to all bus traffic
- Graphically show bus loading by category

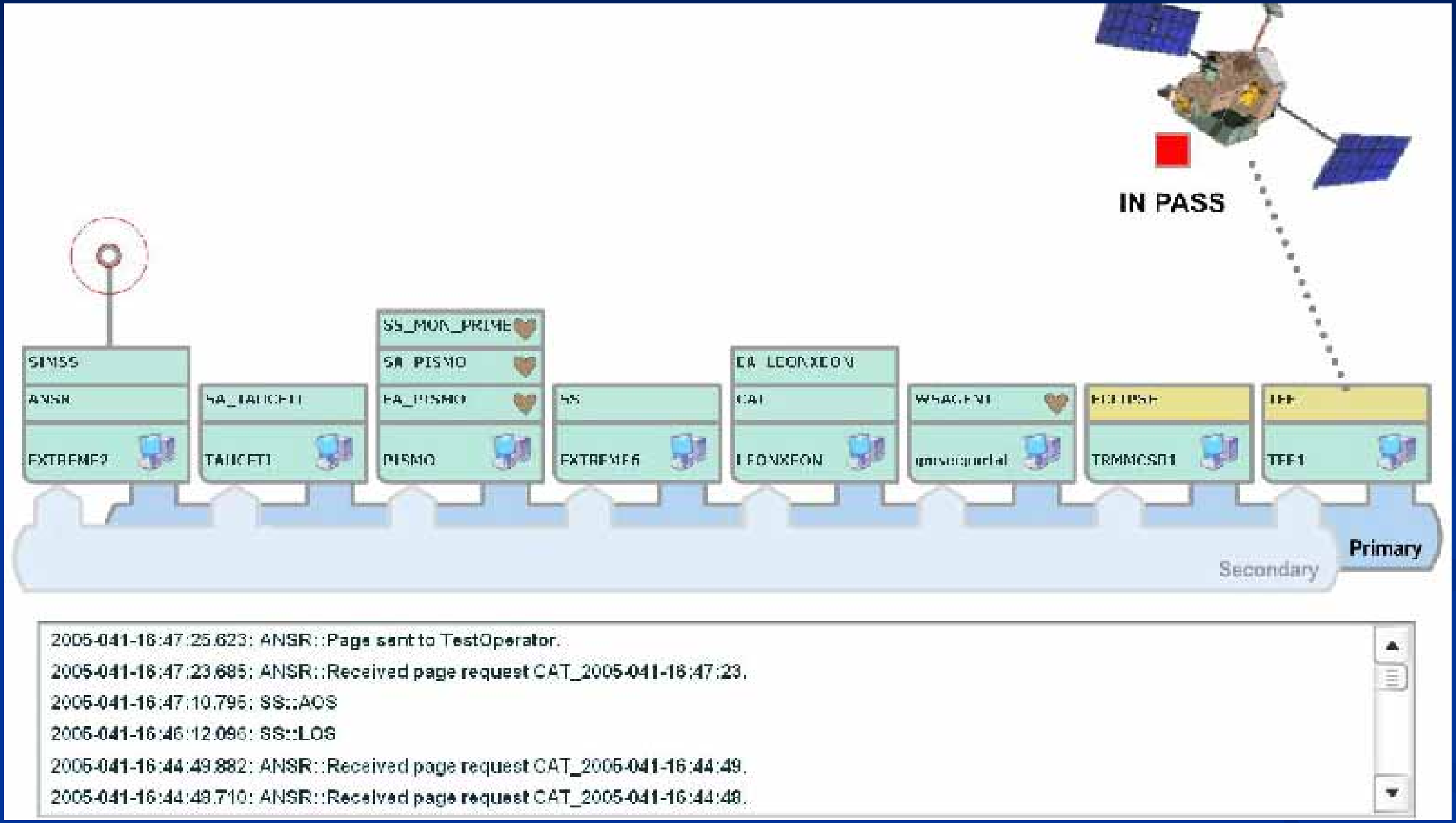
Configuration Display

- Monitor heartbeat messages and selective text messages
- Update graphical configuration display in real-time

Failovers

- Bus and component failover supported by middleware
- Socket connections automatically re-established transparent to component or user

Dynamic Configuration Display



```
2005-041-16:47:25.623: ANSR::Page sent to TestOperator.
2005-041-16:47:23.685: ANSR::Received page request CAT_2005-041-16:47:23.
2005-041-16:47:10.796: SS::AOS
2005-041-16:46:12.096: SS::LOS
2005-041-16:44:49.882: ANSR::Received page request CAT_2005-041-16:44:49.
2005-041-16:44:48.710: ANSR::Received page request CAT_2005-041-16:44:48.
```

Tool subscribes to event messages and heartbeats. Different event messages trigger updates to the display to show start of data flows, pager notifications, software and processor failures, etc. Sound effects for key alarm conditions, failovers, etc. All done with message subscriptions, no integration directly with other components.

Observed Benefits

1. Significant reduction in integration time
2. Components added/upgraded without impacting existing system
3. Ideal for using multiple small distributed development teams and vendors
4. New concepts emerging for small independent components that integrate with the bus and provide immediate benefits
5. Missions more willing to adopt the approach if "old favorite" components can still be used
6. Some vendors see message compliance as a way to enter what had appeared to be a closed marketplace
7. Standard message approach provides collaboration possibilities with other organizations

Return on Investment

- Middleware costs are about $75K per mission
- Costs are saved through lower integration costs, more available tools, increased reuse
- It may not be just about development cost!
  - Systems can be built faster
  - New capabilities are enabled
  - Automation is more powerful, easier to define and expand over time
  - Upgrade approaches are more flexible
  - Do not need to learn details of major components in order to add new independent components.

Final Comments

- The pub/sub component-based architecture is not specific to our domain area, but it does have features which directly aid the challenges faced by our types of missions.
- Other NASA Centers, the Exploration Initiative, commercial satellite operators and other government groups are all evaluating or moving towards GMSEC-like architectures for their future systems.
- GSFC's open approach to collaboration is beginning to bring more business to GSFC as we are recognized as the experts and leaders in the area of new mission data system architectures.